

APPLICATION FOR UNITED STATES LETTERS PATENT

For

DELIVERY OF MULTIMEDIA DESCRIPTIONS USING ACCESS UNITS

Inventors:

Ali Tabatabai

Toby Walker

Visharam Mohammed

Hawley K. Rising III

Prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP

12400 Wilshire Boulevard
Los Angeles, CA 90025-1026
(408) 720-8598

Attorney's Docket No.: 080398.P433

"Express Mail" mailing label number: EL867652 178US

Date of Deposit: October 22, 2001

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Commissioner for Patents, Washington, D. C. 20231

Mara E. Brown
(Typed or printed name of person mailing paper or fee)

Mara E. Brown
(Signature of person mailing paper or fee)

10/22/01
(Date signed)

DELIVERY OF MULTIMEDIA DESCRIPTIONS USING ACCESS UNITS

RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application Nos. 60/272,729, filed March 1, 2001, 60/304,864 filed July 11, 2001, 60/260,949, filed January 10, 2001, and 60/242,378, filed October 20, 2000, the above applications herein incorporated by reference.

FIELD OF THE INVENTION

[0002] This invention relates generally to transmitting data. More specifically, the invention relates to mechanisms and techniques that enable data related to multimedia content descriptions to be encoded and transmitted.

NOTICE OF COPYRIGHT

[0003] A portion of the disclosure of this patent document contains material that is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever. The following notice applies to the software and data as described below and in the drawings hereto: Copyright © 2000, Sony Electronics, Inc., All Rights Reserved.

BACKGROUND OF THE INVENTION

[0004] Digital multimedia information is becoming widely distributed through broadcast transmission, such as digital television signals, and interactive transmission, such as the Internet. The information may be in still images, audio feeds, or video data streams. The Moving Picture Experts Group (MPEG) has promulgated a Multimedia Content Description Interface, commonly referred to as MPEG-7, to standardize the description of multimedia information when it is transmitted from a system that generates the information to a system that uses the information.

[0005] MPEG-7 defines a generic structure for describing multimedia content. For example, the structure for a standard movie could include scenes, shots within scenes, titles for scenes, and time, color, shape, motion, and audio feature information for shots. The corresponding description would contain a "descriptor" component that describes features of the content, such as color, shape, motion, frequency, title, etc., and a "description scheme" component that describes relationships among two or more descriptors, e.g., a shot description scheme that relates together the features of a shot. For example, Fourier descriptors and polygon vertices are representations, or descriptors for a shape feature. A description scheme can also describe the relationship among other description schemes and between description schemes and descriptors, e.g., a scene description scheme that relates the different shots in a scene and relates the title of the scene to the shots.

[0006] The structure and format of an MPEG-7 content description is defined by a schema using a Description Definition Language (DDL), which is designed to define descriptors and description schemes. For each descriptor, the schema specifies the

syntax and semantics of the corresponding feature. For each description scheme, the schema specifies the structure and semantics of the relationships among its children components, which are descriptors and description schemes. The DDL for MPEG-7 multimedia content is based on the XML (extensible markup language) and the XML Schema standards. The descriptors, description schemes, semantics, syntax, and structures of the content description are represented with XML elements and XML attributes. Some of the XML elements and attributes may be optional.

[0007] A multimedia content description is encoded in an XML "instance document" that references the appropriate schema and is an instance of the schema defined in DDL; that is, it contains data that adheres to the syntax and semantics defined in the DDL schema. The instance document contains a set of "descriptor values" for the required elements and attributes in the schema and for any necessary optional elements and/or attributes. An instance document is transmitted from a system having or generating the description data, through a communication channel such as a computer network, to another system that will consume the multimedia content description data contained in the instance document.

[0008] When transmitting a multimedia description represented in XML it is possible to send the data in either textual form or to encode the XML data into another binary form, such as the binary format specified for MPEG-7 data, known as "BiM".

[0009] Figure 1 illustrates one example of transmitting a multimedia document. In system 110, content description 112 is a multimedia description, for example a description of a movie. Encoder 114 encodes content description 112 in a format suitable for transmission. After encoding, encoded data stream 116 represents the encoded form

used for transmitting content description 112 to decoder 118 over some communication channel (not shown). Decoder 118 receives encoded data stream 116 and uses it to reconstruct content description 120, which is the same as content description 112. Encoder 114 may use various methods to form the encoded data used to transmit content description 112.

[0010] While various means may be used by encoder 114 to transmit encoded data stream 116 to decoder 118, normally the description data must be sent as an encoded data stream over a communication channel, such as a network communication channel using a communications protocol like TCP/IP. Although compression can reduce transmission time by decreasing the size of the encoded stream, if the description is large, transmitting the entire description over a network can still take too much time. Rather than sending the entire description, the encoder may send parts of the description to the decoder. The method for determining which parts to send is not the subject of this invention and is application dependent. Herein, the term "description fragment" means a part of a description.

[0011] The content description 112 may be either static or dynamic. A static description is one that does not change during the duration of communication between encoder 114 and decoder 118. A dynamic description is one in which the description data 112 is changed during the duration of the communication between encoder 114 and decoder 118. This change must be reflected in the description data 120. Description data 112 is changed when data pertaining to the multimedia content is added, deleted, or changed. For example, if the description pertains to a scene being captured by a camera the description may change if an object appears or disappears from the scene. Additionally, if the description is about a television program, portions of the data may be

updated, e.g. the broadcast time of the program may changes, and such changes sent from the encoder to decoder. When the description is dynamic it is more efficient to send only updates to the encoder rather the resending the entire changed description.

[0012] Furthermore, in a distributed environment, description data is spread across multiple hosts and each host contains only a part of a complete description. If all description fragments (i.e., parts of the description) are streamed from a single encoder, then the encoder spends a significant amount of time and resources in gathering all the description fragments.

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219

SUMMARY OF THE INVENTION

[0013] One embodiment is a method comprising the following process: form an access unit comprising a fragment update, where the fragment update comprises a fragment update command; form an encoded data stream from the access unit.

BRIEF DESCRIPTION OF THE DRAWINGS

- [0014] Figure 1 is prior art and illustrates an example of transmitting a description from an encoder to a decoder.
- [0015] Figure 2 illustrates an example of a content description tree and its fragments.
- [0016] Figure 3 illustrates an example of an access unit.
- [0017] Figure 4 illustrates an example of a fragment update.
- [0018] Figure 5a is a flow diagram showing blocks, which when implemented by an encoder send an encoded data stream to decoder 510.
- [0019] Figure 5b is a flow diagram showing blocks, which when implemented by a decoder provide instance document 116 with both text and binary encoding.
- [0020] Figure 6 is a flow diagram showing one embodiment of the changes made to BiM syntax.
- [0021] Figure 7 illustrates one example of BiM syntax.
- [0022] Figure 8 illustrates an example of a distributed MPEG-7 description.
- [0023] Figure 9 illustrates one example of a fragment reference.
- [0024] Figure 10 illustrates one example of internal and external pointers.
- [0025] Figure 11 illustrates one example of a computer system that implements the principles of the invention.

DETAILED DESCRIPTION

[0026] In the following description, numerous specific details such as specific materials, processing parameters, processing blocks, etc., are set forth in order to provide a though understanding of the invention. One skilled in the art will recognize that these details need not be specifically adhered to in order to practice the claimed invention. In other instances, well known processing blocks, materials, etc., are not set forth in order not to obscure the invention. Furthermore, different embodiments of this invention may use different syntaxes for encoding substantively the same information. For example, the use of different names, attributes, elements to form alternative embodiments of this invention will be will be obvious to one well-versed in the art.

[0027] This invention provides an efficient method to deliver MPEG-7 description data from an encoder to a decoder using access units, a method for configuring a system to receive access units, and a method for using access units for communication of distributed multimedia content description. Furthermore, the invention is not limited to MPEG-7 multimedia description and may also be applied to any structured data represented using XML.

[0028] When sending a description from a decoder to an encoder, the description is usually divided into pieces that are sent to a decoder one after another. In the following, "access unit" is used to refer to the smallest unit for transmitting a part of a description from an encoder to a decoder across a communication channel or inside a stream. The term is chosen because the access unit is the smallest unit for a decoder accessing a description. When transmitting a description the encoder divides the description into

fragments, encodes these as access units, and sends access units to the decoder where they are used to reconstruct the description.

[0029] The description tree, defined in detail later, is a representation of hierarchical structure of elements, attributes, text values and so on, that make up an instance document. A "fragment" is an arbitrarily connected component of the description tree that is not necessarily complete with respect to a schema definition, such as MPEG-7 DDL. In other words, a fragment is a sub-tree of the description tree.

[0030] An encoded transmission format typically provides information pertaining to the method of encoding transmitted data. For example, encoder 114 sends information about the encoded data stream 116 to the decoder. Such information may be sent over the same communication channel or stream (e.g. "in-band") or "out-of-band"; that is, on a separate communication channel from that used for sending the encoded description stream. An encoder provides information about the resources needed for decoding so that a decoder can correctly decode an instance document. Information about resources needed for decoding allows a system to determine if the needed resources are present and available, and if not then to send a command to stop transmission. One method of providing resource information is to send a global element with information (e.g., MPEG7_DecoderConfig) at the beginning of a session.

[0031] When the description data is transmitted or streamed, for example when a description is broadcast, a decoder needs to determine a correct beginning point within encoded data stream. To do so the encoded data typically has synchronization points. Synchronization points are points in the encoded data stream where a decoder can form an initial state, start decoding, and then properly decode subsequent encoded data.

[0032] Synchronization points enable random access to the encoded stream of description data. This is analogous to the "I-Frame" used in video compression standard, such as the MPEG-1 and MPEG-2 standards. Such access points are associated with a partial description, or sub-tree, where a decoder can access only a part of the entire description. Such sub-trees must be semantically meaningful points. That is, they must be sub-tree that is an instance of a descriptor, description scheme, or datatype defined by the MPEG-7 standard schema. Such semantically meaningful points may be defined as MPEG7Unit.

[0033] Another example of a synchronization point is a reset point that contains a fragment that forms a complete description with respect to the schema, for example MPEG7Main, a complete, independent, descriptive entity.

[0034] In some cases a multimedia content description is distributed over multiple servers. In one embodiment, in order to support rapid and efficient transmission of multimedia work stored on multiple servers the encoder transmits the location of description fragments instead of transmitting the description fragments themselves to the decoder. The decoder can then directly retrieve the description fragments from other servers.

[0035] Figure 2 illustrates an example of a content description tree and its fragments. Content description tree 200 represents a multimedia content description and is part of the data in instance document 112. The structure of the description is determined from the hierarchical structure of the XML elements and attributes used to represent the multimedia description. The details of this structure are specified in the DDL schema for the description. As shown in Figure 2, a description can be divided into a set of fragments

that can be "glued" back together to reconstruct the original description tree. For example, encoder 114 divides content description tree 200 into a set of fragments 202. Fragments 202 1-3 are subtrees of content description tree 200.

[0036] One method for transmitting a description is for the decoder to send it as a series of fragments, which are transmitted and received by the decoder, which then recombines them to reconstruct the complete description. Because each fragment 202 is sent separately, there must be information provided for each fragment 202 that identifies its location within content description tree 200. Fragments 202- 2 and 3 have information relating their position to fragment 202-1. Consequently, even though fragments 202 are sent separately, a corresponding decoder can reassemble fragments 202 into content description tree 200. The encoder does so by sending the fragment with its associated information as an access unit.

[0037] Figure 3 is an example of an access unit. Henceforth, the abbreviation AU refers to an "access unit". Access unit 300 is composed of an optional digital signature 302 and one or more fragment updates 304. One embodiment of access unit 300 is an MPEG-7 system datatype to which timing information is attached. Access units typically do not carry the same timing information. The timing information is used for, in one example, to synchronize a transmitter (not shown) and a receiver (not shown) or to coordinate the delivery of the description with its associated multimedia content. In many communication protocols, the packets in the transport layer packets carry timing information, such as delivery times.

[0038] Figure 4 is one example of a fragment update. Fragment update 400 is composed of three parts. Navigation part 402 points to a location in content description

tree 200 where update command 404 should execute. In other words, it specifies how to navigate to the node in the description where the command is to be applied. Update command 304 specifies the type of update to execute, for example to add, delete, or replace a fragment. Payload 406 provides the value of fragment 202 to be added or replaced but is not present when a delete command is sent.

[0039] The inclusion of an update command inside of an access unit allows an encoder to send either static or dynamic descriptions. When the description being transmitted is static, the encoder can divide the description into fragments, which will typically consists of add commands to build up the description. When the description is dynamic, an encoder can send a series one or more access units that contain update commands to effect any changes on the source description at the destination decoder.

[0040] Figure 5a is a flow diagram of blocks, which when implemented by an encoder and a decoder encode an access unit. In block 502 the decoder calculate the description tree fragments that need to be sent or updated. The process for doing this is encoder and application dependent and is not discussed here. In block 504 the decoder determines the navigation paths for the updated description. These are the paths to the root nodes of the description sub-trees that are to be updated. In block 506 the encoder gathers together all of the information needed to send the update and fragment in an access unit. For example, the decoder may represent the information in an access using an XML syntax discussed later or in a binary encoding. In block 508 the encoder sends the access unit information to the decoder in an encoded data stream.

[0041] Figure 5b is a flow diagram of blocks, which when implemented by an encoder and a decoder decode an access unit. The decoder performs the processing

depicted in Figure 5b. In block 512 the decoder receives an access unit from the encoded data stream. In block 514 the information in the AU is decoded. For example, this may involve decoding a binary form of the access unit or parsing a textual XML representation of the access unit. In block 516, the navigation part is used to determine the affected node in the current description maintained by the decoder. In MPEG-7, for example, the navigation is expressed as an XPath expression that locates the root of the sub-tree that the decoder will update.

[0042] In block 518, complete the access units. Each access unit has a set of update commands and an indication of whether or not the access unit is applied serially or in parallel. In this block the update command is examined in order to determine which operation is specified: an add, delete, or replace command. If it is a delete command the sub-tree is deleted by the decoder in block 520. In block 522 the fragment in the update command is added to a designated node. In block 524 the designated fragment is replaced by the fragment value in the update command.

[0043] Figure 6 is a flow diagram showing one embodiment of block 516 when the navigation path is represented as an Xpath expression (or equivalent path representation). Determine if there is a '/', block 600. If there is no '/' then absolute addressing is not used, block 602.

[0044] Determine if the '/' is followed by a root node, block 604. If the '/' is followed by a root node then conform to mode '00' of BiM navigation command, block 606.

[0045] Determine if the '/' is followed by a top level node, step 608. If the '/' is followed by a top level node then conform to mode '01' of BiM navigation command, block 610.

[0046] Determine if there is a '.', block 612. If there is a '.' Then context node is indicated, block 614.

[0047] Determine if there is a '..', block 616. If there is a '..' then parent of a context node is indicated, block 618.

[0048] Determine if '/' is preceded by '..', block 620. If '/' is preceded by '..' then conform to relative addressing mode '11' of BiM navigation command, block 622.

[0049] In order to decode access units, a decoder needs an indication of where in an encoded data stream to begin decoding. Synchronization points are one example of points at which a decoder begins decoding and produces correct output. In one embodiment, a synchronization point is a random access point associated with a description fragment where a decoder can decode a part of the description and obtain a semantically correct portion of the description. As explain earlier, a semantically correct fragment is one that corresponds to a descriptor, description scheme, or datatype defined in the MPEG-7 schema.

[0050] One embodiment of a synchronization point is a reset. In one embodiment, a reset point is implemented using a "reset" update command. The access unit containing a reset command contains a fragment, or fragments, that form a complete description tree with respect to the schema. Such an access unit serves as a reset point for a decoder as it resets the state of the description decoding process. One example of a reset point is content description tree 200 in Figure 2.

[0051] Yet another embodiment of a synchronization point is a fragment. Examples of fragments include fragments 202 1-3. A fragment is an arbitrarily connected component of the description tree that does not need to be complete with respect to any XML schema.

[0052] In the encoded data stream, a special sequence can be used to indicate the occurrence of a synchronization point. Figure 7 illustrates one example of a synchronization code that is possible for MPEG-7 BiM syntax. The twenty-four bit pattern 700, followed by a three bit pattern 702 indicates that a synchronization point is present. Three-bit pattern 702 indicates MPEG7Unit or MPEG7Main. In order to indicate a fragment, modify three bit pattern 702 to reflect signaling of a fragment.

[0053] In one embodiment, signal synchronization points at the AU level by defining an attribute for MPEG7_AccessUnit called "sync_point," enumerating the three possibilities.

[0054] Figure 8 illustrates a distributed MPEG-7 description. Content description tree 800 is distributed among hosts 802. Host 802-1 stores fragment 804-1, host 802-2 stores fragment 804-2, host 802-3 stores fragment 804-3, and host 802-4 stores fragment 804-4. In the prior art, in order to transmit content description tree 800, all fragments are transmitted from a single encoder.

[0055] Rather than transmit all the fragments values in an access unit, a single host, for example host 802-3 with description data in fragment 804-3, transmits fragment 804-3 with pointers, or fragment references, to audio description data in fragment 804-1 and video description data in fragment 804-2.

[0056] Figure 9 illustrates one possible implementation of a fragment reference with access units, which is an extension and modification of the implementation of access units illustrated in Figure 3. Access unit 900 contains fragment update 902. Rather than a payload in fragment update 902, as would be found in the prior art, the invention shows fragment reference 904. Fragment reference 904 is a reference that identifies an accessible resource containing the fragment.

[0057] In one embodiment, any resource designated by a uniform resource identifier (URI) can be available as the source for a fragment value. For example, a URI of the form "http://..." indicates Hypertext transfer Protocol (HTTP) form of a URI and points to referenced fragment 906. In one example, the source is a node identified in some MPEG-7 description, though the source is not necessarily in the same description document as the description document being transmitted.

[0058] In another embodiment, the reference is to a previously sent node that is used as a fragment value. This allows transmission of identical subtrees. The reference is typically in XPath.

[0059] In another embodiment the reference is to a node in the current description tree. In one example the reference uses an XML IDREF.

[0060] Any resource that represents an MPEG-7 description whose type is appropriate for the fragment value can be referenced. In one embodiment the referenced fragment comes from the same description tree as the one being transmitted. In another embodiment the referenced fragment comes from a different description tree as the one being transmitted. Fragment reference 904 preferably resolves to a node that is the root of a description reference.

[0061] There are several embodiments in which a decoder (not shown) accesses the referenced fragment. In one embodiment the decoder receives the referenced fragment through an MPEG-7 transmission system whereby the fragment is transmitted as a partial description using access units.

[0062] In another embodiment the decoder stores fragment values. The decoder compares referenced fragments to stored fragments and if a referenced fragment is stored then the decoder obtains the fragment from storage; otherwise, the decoder retrieves the fragment from the referenced location.

[0063] In another embodiment the decoder uses an HTML address to access an XML document from a network (not shown).

[0064] In another embodiment the decoder accesses fragment references only when needed, rather than when first received. Such an embodiment allows the decoder to have greater control over the timing and reception of description data. For example, if the decoder does not need a particular fragment reference, it may never access and retrieve that fragment reference.

[0065] A fragment that is referenced may have links that point to nodes outside of the set of nodes inside the fragment's sub-tree. When a fragment is referenced the decoder must determine how such links are to be correctly resolved.

[0066] Internal links point to nodes inside the same document as the fragment. An internal link can lose its meaning if the fragment is removed from the original document. An external link is points to a node in another document. One example of external links are links implemented in XML by a URI reference. External pointers within a fragment do not change their meaning if they are removed from the original document.

[0067] Figure 10 illustrates one example of internal and external links. Doc 1000 has node 1002 with an internal pointer to node 1004. Nodes 1002 and 1004 have identifiers unique to doc 1000, for example nodes "a" and "b", respectively. Node 1002 references node identified as "b", which is node 1004.

[0068] Suppose that a decoder references a fragment sub-tree rooted at node 1002 in Doc 1 and another fragment 1008 in Doc 2. Further suppose that a the sub-tree rooted at node 1012 in Doc 1010 is to be replaced with a fragment that is referenced, or a referenced fragment.

[0069] If the update for fragment 1002 is applied to document 1010 then the result is shown as document 1016. Note that in the updated sub-tree 1018 the internal link shown at node 1020 points to the node B in document 3 rather than the original target of the internal link, which was node 1004 in document 1.

[0070] In another embodiment, when fragment 1008 is used to update node 1012 of document 1010 the result is illustrated as document 1022. The updated sub-tree at node 1024 still points to the node B in document 1010 as in the original document 1006.

[0071] In another embodiment, node 1002 is translated into an external link to node 1004 (node 'B' of doc1).

[0072] In yet another embodiment, within a fragment update an encoder encodes a fragment reference with a binary encoding format and a command with a text-based encoding format.

[0073] In yet another embodiment, a fragment update comprises both a payload and a fragment reference.

[0074] Referencing fragments makes, in one embodiment, distributing MPEG-7 description data at the system level transparent to the description itself. Fragment references enable distribution of large descriptions across multiple servers.

[0075] Referencing fragments also allows sharing and replicating description references that are used in multiple descriptions.

[0076] Transmission size is compressed if there are common references within a description tree. An encoder transmits a reference once and then sends a pointer to that reference in subsequent access units.

[0077] In one embodiment the DDL definition for decoder configuration information, named DecoderConfig here is:

```
<element name= "DecoderConfig" type= "mpeg7:DecoderConfigType"/>
<complexType name= "DecoderConfigType">
  <choice>
    <element name= "Predefined">
      <simpleType>
        <restriction base= "positiveInteger">
          <minInclusive value= "1"/>
          <maxInclusive value= "255"/>
        </restriction>
      </simpleType>
    </element>
    <element name= "SchemaReference" type= "uriReference" minOccurs=
      "1" maxOccurs= "unbounded"/>
  </choice>
  <element ref= "MPEG7_AccessUnit" minOccurs= "0" maxOccurs= "1"/>
  <element ref= "Profile" minOccurs= "0" maxOccurs= "1"/>
</complexType>
```

Predefined indicates whether or not to use a predefined configuration. In one embodiment a list of uniform resource identifiers (URIs) follows. In another embodiment the number of URIs is zero and a well-known set of schemas is used instead. SchemaReference is an URI to reference the schema. AccessUnit provides the 'reset'

and initial state description, supplied inline, that forms the basis of the reset and initial states.

[0078] In one embodiment, the DDL representation definition for an access unit, called MPEG7_AccessUnit here, is:

```
<complexType name="MPEG7_AccessUnit">
  <element name="update" type="updateCommand" minOccurs="1"
    maxOccurs="unbounded"/>
  <attribute name="executionOrder">
    <simpleType base="string">
      <enumeration value="sequential"/>
      <enumeration value="parallel"/>
    </simpleType>
  </attribute>
</complexType>

<complexType name="updateCommand">
  <element name="command" type="updateType"/>
  <element name="location" type="updateLocation"/>
  <element name="value" type="updateValue"/>
</complexType>

<simpleType name="updateType" base="string">
  <enumeration value="add"/>
  <enumeration value="delete"/>
  <enumeration value="change"/>
  <enumeration value="reset"/>
</simpleType>

<complexType name="updateLocation">
  <attribute name="fragmentID" type="uriReference"/>
  <attribute name="position" type="boolean" use="default" value="1"/>
</complexType>
```

```

<complexType name="updateValue" minOccurs="0" maxOccurs="1">
  <choice>
    <element name="binaryValue" type="binaryUpdate"/>
    <element name="textualValue" type="textualUpdate"/>
    <element ref="resetContentDescription"/>
  </choice>
  <attribute name="schemaIndex" type="integer"/>
</complexType>

<simpleType name="listOfByte" base="Byte" derivedBy="list">

<complexType name="VectorB" base="listOfByte" derivedBy="extension">
  <attribute ref="mpeg7:dim"/>
</complexType>

<complexType name="binaryValue">
  <element name="value" type="VectorB"/>
  <!-- Using "NOTATION" will point to which decoder to use for nonXML
content.
and it is optional.-->
  <attribute name="binaryType" type="NOTATION" use="optional"/>
</complexType>

<complexType name="textualValue">
  <any namespace="##other" minOccurs="1" maxOccurs="unbounded"
processContent="skip"/>
</complexType>

```

UpdateType indicates the type of update command used, for example add, delete, change and reset. In one embodiment updateType is update command 404. In one embodiment the reset command replaces the whole content description tree with a predefined or referenced content description tree. updateLocation uses a subset of the Xpath language to indicate where the update command executes. In one embodiment updateLocation is navigation part 402. In one embodiment there is a flag indicating whether an addition occurs before or after the identified location. updateValue indicates the value to be added or replaced. In one embodiment updateValue is payload 406. In one embodiment the

update value is specified either as a binary or textual value using the binaryValue or textualValue datatypes respectively.

[0079] In another embodiment, DDL representation syntax for a textually-encoded access unit is:

```
<element name= "AccessUnit type="mpeg7:AccessUnitType"/>
<complexType name= "AccessUnitType">
  <sequence>
    <element name= "FragmentRef" type= "mpeg7:FragmentUpdateType"
      minOccurs= "0" maxOccurs= "unbounded"/>
  </sequence>
  <attribute name= "Order" use="optional"/>
  <simpleType>
    <restriction base= "string">
      <enumeration value= "seq"/>
      <enumeration value= "par"/>
    </restriction>
  </SimpleType>
</attribute>
<attribute name= "auID" type= "NcName">
</complexType>
```

In one embodiment a fragment reference associates the sub-tree included in the same AU with its child sub-tree included in another AU. In another embodiment a fragment update is used for dynamic building up a content description tree. In one embodiment order is used to signal as to whether the update order should be preserved between a sequence of FragmentUpdates.

[0080] In another embodiment, DDL representation syntax for the syntax for accessing units allows either fragments or fragment references as follows:

```
<complexType name="FragmentPayloadType">
  <complexContent>
    <choice>
      <element name="Payload" type="mpeg7:Mpeg7RootType"/>
      <element name="Reference" type="mpeg7:ReferenceType"/>
    </choice>
    <attribute name="PartialInstantiation" type="boolean">
```

use="default" value="false"/>
</complexContent>
</complexType>

[0081] **Payload** contains the value of a description fragment to be updated.

Reference contains a URI reference to fragment that can be used as the source for the value of the update. **PartialInstantiation** is an attribute used to signal that one or several elements of the fragment have empty content. Payload contains an MPEG-7 root unit, which is either a complete description or a single descriptor or description instance.

[0082] In one embodiment, **ConnectionPoint** indicates a node of the sub-tree to which a child sub-tree should be connected. In one embodiment **FragmentLocation** indicates a child sub-tree included in another AU.

[0083] Figure 11 illustrates an embodiment of computer system 1100 that implements the principles of the invention. System 1100 includes a stand-alone or portable computing device. Computer system comprises processor 1102, storage device 1104, and MPEG encoder 1106, all coupled together with interconnect 1108, such as a bus or a point-to-point pointer. In addition, a number of user input/output devices (not shown) couple to system 1100.

[0084] Processor 1102 represents a central processing unit of any type of architecture (e.g., Intel architecture or Sun Microsystems architecture), or hybrid architecture. In addition, processor 1102 could be implemented in one or more chips. Storage device 1104 represents one or more mechanisms for storing data such as content description trees or fragments. Storage device 1104 represents read-only memory (ROM), random access memory (RAM), magnetic disk storage media, optical storage media, flash memory devices, and/or other machine-readable media. Interconnect 1108 represents one

or more buses (e.g., accelerated graphics port bus, peripheral component interconnect bus, industry standard architecture bus, X-Bus, video electronics standards association related to buses, etc.) and bridges (also termed bus controllers).

[0085] While this embodiment is described in relation to a single processor computer system, the invention could be implemented in a multi-processor computer system. In addition to other devices, one or more networks 1110 may be present. Network 1110 represents one or more network connections for transmitting data over a machine-readable media. The invention could also be implemented on multiple computers connected via such a network.

[0086] Storage device 1104 has stored therein data 1112 and program instructions 1114 (e.g., software, computer programs, etc.). Data 1112 represents data stored in one or more of the formats described herein. Program instructions 1114 represent the necessary code for performing any and/or all of the techniques described with reference to Figures 2-10. It will be recognized by one of ordinary skill in the art that storage device 1104 preferably contains additional software (not shown), which is not necessary to understanding the invention.

[0087] System 1100 also includes encoder 1106 coupled to processor 1102 and storage device 1104. Encoder 1106 is not necessarily based in hardware, it may also be stored as instructions 1114 in storage device 1104 and executed by processor 1102. Encoder 1106 is used for encoding control signals and/or microcode entry points sent by processor 1102. In response to these control signals and/or microcode entry points, encoder 1106 encodes the control signals so that they may be transmitted across network

1110 to system 1116 where processor 1118 and decoder 1120 will perform the appropriate operations.

[0088] The particular methods of the invention have been described in terms of computer software with reference to a series of flow diagrams. The methods to be performed by a computer constitute computer programs made up of computer-executable instructions illustrated as blocks (acts). Describing the methods by reference to a flow diagram enables one skilled in the art to develop such programs including such instructions to carry out the methods on suitably configured computers (the processing unit of the computer executing the instructions from computer-readable media). The computer-executable instructions may be written in a computer programming language or may be embodied in firmware logic. If written in a programming language conforming to a recognized standard, such instructions can be executed on a variety of hardware platforms and for interface to a variety of operating systems. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein. Furthermore, it is common in the art to speak of software, in one form or another (e.g., program, procedure, process, application, module, logic...), as taking an action or causing a result. Such expressions are merely a shorthand way of saying that execution of the software by a computer causes the processor of the computer to perform an action or to produce a result. It will be appreciated that more or fewer processes may be incorporated into the methods as described above without departing from the scope of the invention, and that no particular order is implied by the arrangement of blocks shown and described herein.

